International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-8, June 2020

New Parallel Technics for GPU, Fast SURF Algorithm

Hicham Hassnaoui, Aisha Sahel, Abdelmajid Badri

Abstract: ***Computer vision algorithms, especially real-time tasks, require intensive computation and reduced time. That's why many algorithms are developed for interest point detection and description. For instance, SURF (Speeded Up Robust Feature) is extensively adopted in tracking or detecting forms and objects. SURF algorithm remains complex and massive in term of computation. So, it's a challenge for real time usage on CPU. In this paper we propose a fast SURF parallel computation algorithm designed for Graphics-Processing-Unit (GPU). We describe different states of the algorithm in detail, using several optimizations. Our method can improve significantly the original application by reducing the computation time. Thus, it presents a good performance for real-time processing.***

Keywords: **Computer vision, GPU, parallel computation, SURF, Tracking.**

I. INTRODUCTION

Visual recognition is based on feature points detection as the essential step in the most algorithms, e.g. object tracking, stereo image rectification, facial recognition etc. these algorithms characterized by invariance to: rotation, scale change, affine transformation and illumination changes.

The first works, found in the literature, which take into account rotational invariance are those of Harris [1] who proposed a second-moment matrix method. Lindeberg [5] introduced a scale-invariance method using a Hessian matrix determinant. Lowe [4] proposed a new algorithm called SIFT combining scale and rotation invariance. SIFT is considered powerful at the scale invariance but in terms of speed remains limited, which deprives it from being used for real-time applications. An idea used for improving SIFT is proposed by Bay [17] called SURF (speeded up robust features). Now SURF algorithm is widely used in many applications as tracking, image registration, video segmentation etc. Despite these performances, the implementation of this algorithm on CPUs remains unable to meet requirements given for the sequential architecture of CPU processors.

Revised Manuscript Received on June 30, 2020.
* Correspondence Author
Hicham Hassnaoui*, Student, Department of Computer vision, Faculty of Sciences and Techniques, Mohammedia (FSTM) of UH2C (University Hassan II - Casablanca) – Morocco. E-mail: hicham.hassnaoui@gmail.com
Aicha Sahel, Professor, Department of Electronics and Image Processing, Faculty of Sciences and Techniques (of Mohammedia), of UH2C (University Hassan II - Casablanca) – Morocco. E-mail: sahel_ai@yahoo.fr
Abdelmajid Badri, Professor, Department of Electrical Engineering, University of Hassan II of Casablanca. E-mail: abdelmajid_badri@yahoo.fr

To improve processing speed of SURF, many solutions are proposed using different platforms such as FPGA, multi-core processor or graphical processor (GPU). Fan [8], Chen [6] work on FPGA architecture to implement parallel SURF. Wilson [3] proposed another real time solution for FPGA. Chao [2] proposed a novel rate control approach in video compression. Timothy [7] proposed a method of parallelization of the algorithm to implement on GPU. First Timothy used a prefix sum 2D approach to calculate Integral Image as first step of the algorithm then he uses a box filter optimization to calculate a Hessian determinant in different scales (in parallel). In this paper, we combine some previous methods and we propose a new parallelization method, as described in detail in section III, to achieve a good optimization to improve processing speed, using OpenCL implementation on GPU.

II. INTEREST POINTS DETECTION BY SURF

The SURF algorithm consists of two phases, feature point detection and descriptors calculation. Feature point detection achieved in four steps and then two steps for descriptors calculation as shown in (fig.1).

A. Integral Image

Based on the input image (or video frame) at the first step, we compute the called Integral Image (I.I.) given by equation (1).

$$I_{\Sigma}(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \tag{1}$$

I.I. is used to accelerate the computation, indeed if we take any rectangular zone (fig.2) only three additions needed to get the sum of all pixel's intensities inside this zone. Hence the computation time is the same whatever the size.

$$\sum = I_{\Sigma}(A) + I_{\Sigma}(D) - I_{\Sigma}(B) + I_{\Sigma}(C) \tag{2}$$

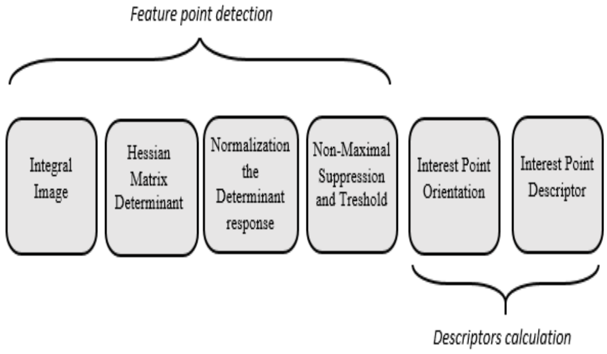

Fig1. Stages of SURF Algorithm

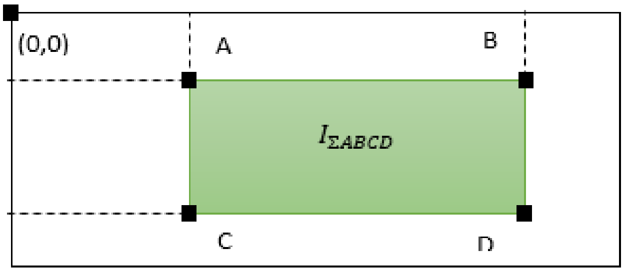

Fig2. Illustration of Integral Image in square region

B. Hessian Matrix Determinant

At the second step of the algorithm, we compute a Hessian matrix as a convolution of the image with second order partial derivative of Gaussian in three different directions x, y and xy. The determinant of Hessian matrix is computed by equation (3).

$$D(x,y) = det\begin{bmatrix} Gxx & Gxy \\ Gxy & Gyy \end{bmatrix} = det\begin{bmatrix} \delta^2 f/_{\delta x^2} & \delta^2 f/_{\delta x \delta y} \\ \delta^2 f/_{\delta x \delta y} & \delta^2 f/_{\delta y^2} \end{bmatrix} \quad (3)$$

Gaussian kernels are continuous. Therefore, a discretized approximation of these kernels is used [17]. Then the Hessian determinant is given by equation (4):

$$D(x,y) = G_{xx}\,.\,G_{yy} - w^2.G_{xy}{}^2 \quad (4)$$

Here is a correction weight equal to 0.9 [17]. Bay [17] uses blob response map for the storage of the responses according to the scale.

C. Removal of non-maxima and thresholding

In the fourth stage, we apply a non-maximal suppression by finding the determinant maximum value within the 26 neighbors in current, previous and next scale [17]. Afterwards we apply a threshold to keep the strongest interest points only. Up to now the interest points have been extracted, what remains is the assignment of orientations to these points and the computation of descriptors.

In the next stage, the orientation computation is adopted by calculating Haar wavelets response among both x and y axes in circular neighborhood of radius 6σ around the key point [17]. After that, we choose the dominant orientation by calculating the sum of the responses horizontally and vertically. Then, In the last stage, the descriptor is computed, using a circle (or square) region centered on the key point and have the orientation calculated previously.

III. PARALLELISM ANALYSIS OF SURF

A. Integral image parallelization

Computation of integral image is an expensive step; in its calculation we can use 1D prefix sum on rows and then a prefix sum on columns. To reduce computation time Blelloch [14] proposes a 2D parallel prefix sum. He used a two-phase method, one called Up-sweep and a the other Down-sweep, thus construct two pyramids. (Fig 3). When the Down-sweep process is finished, we get in each vertex the sum of all previous leaf values. Blelloch Theorem 1.1 [14]. So, using Blelloch's method and considering an image NxN we can easily prove that the algorithm has a time complexity of: O(2Log(N)).

Feature Detection and nom maximum suppression Bay et al. [17] gave the filter approximation specification for the first scale, but not for the following scales, so some sampling details of the other scales are ambiguous. Timothy et al. [7] introduce four parameters Q1,…,Q4. For the precise layout of the box filter as given in fig.4, these parameters values, with associated σ for each scale, are resumed in table II. So, we will proceed in the same manner like Bay et al. using the same sizes {9, 15, 21, 27…}. Hence, we divide the scale space into octaves. Each octave contains a set of response maps. These maps are a convolution of the Integral Image with gaussian filters of variant size. An octave contains 4 layers. The first one starts with a 9x9 filter, for the smallest scale (σ=1.2) and then we continue increasing layer's size by 6 pixels in the first octave, by 12 in the second octave and by 24 in the 3rd octave etc. Refer to table I.

The scale σ in a layer of size NxN is obtained by the formula: σ=1.2×N/9. We convolve these 24 box filters (8 scales ×3 derivatives), with the input image in parallel and we compute the determinant of Hessian matrix in each location with Equation 4. Thus, constituting a response map with different scales, every response is a vector x= (x, y, σ). An interest point candidate is obtained by the non-maximum suppression in 3×3×3 neighborhoods in 3-dimentional scale space (x, y, σ). The maxima are then interpolated in scale space (x, y, σ) to localize the accurate feature point [17].

Table I: SCALE SPACE CONSTRUCTION

	Layer 1	*Layer 2*	*Layer 3*	*Layer 4*
Octave 1	*9*	*15*	*21*	*27*
Octave 2	*15*	*27*	*39*	*51*
Octave 3	*27*	*51*	*75*	*99*

Table II: BOX FILTER PARAMETERS

Filter size	*Q1*	*Q2*	*Q3*	*Q4*	σ
9	3	5	3	1	1.2
15	5	9	5	1	2
21	7	11	7	3	2.8
27	9	15	9	3	3.6
39	13	21	13	5	5.2
51	17	29	17	5	6.8
75	25	41	25	9	10
99	33	55	33	11	13.2

The computation of Hessian components values for different scales are independent, so we can apply these 24 box filters simultaneously, then we compute the determinant response for 8 scales in parallel. The non-maximum suppression process, by comparing the center point with the other 26 neighborhoods in scale space, can be operated in parallel. Even more we can divide the image into several blocks and thus treat each block independently so in parallel. Finally, all the process can be pipelined (pipeline parallelism). Refer to fig.5.

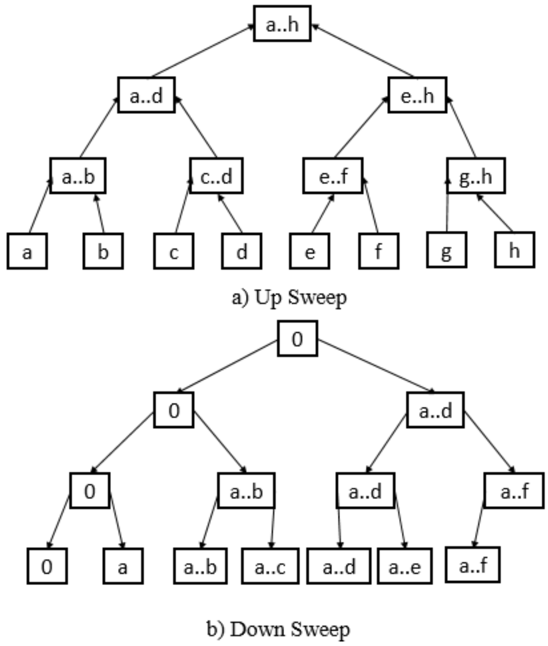

Fig. 3: Blelloch's Two phases parallel prefix sum

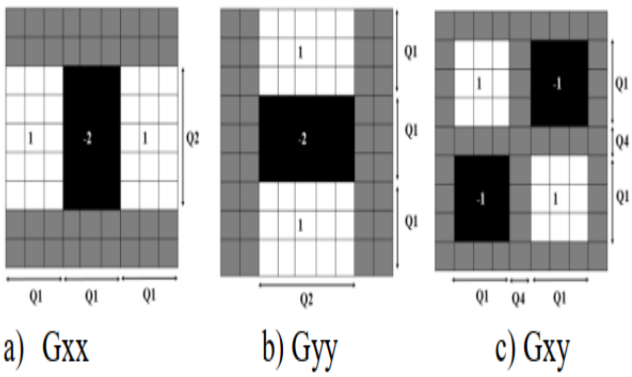

Fig. 4: 9X9 Box filter approximating 2nd order partial derivatives of a Gaussian

IV. IMPLEMENTATION

The OpenCl framework for image processing is given in fig.6. The input image data is copied to GPU memory (Device global memory), by using ND Range adaptively each work item execute the kernel using local memory, to accelerate the access, then we copy back data to CPU (Host memory).

Our OpenCl implementation of SURF consist of the following steps:

1) RBG to gray conversion.
2) Integral image computation (2D prefix sum).
3) Hessian Determinant.
4) Non-maxima supression.
5) Orientation assignement.
6) Descriptor components.

These steps are computed using a total of 23 kernels. In these kernels we used a work groups of 16x16 work items in each work group. The number of work items is the same as the number of pixels, hence we really improve the computation speed.

As a result, after this optimization we implemented our new algorithm with OpenCL in NVIDIA GeForce GT540M, (fig.7, fig.8) on images of different sizes, it gave considerable results.

Indeed, many times faster compared to the original version of SURF.

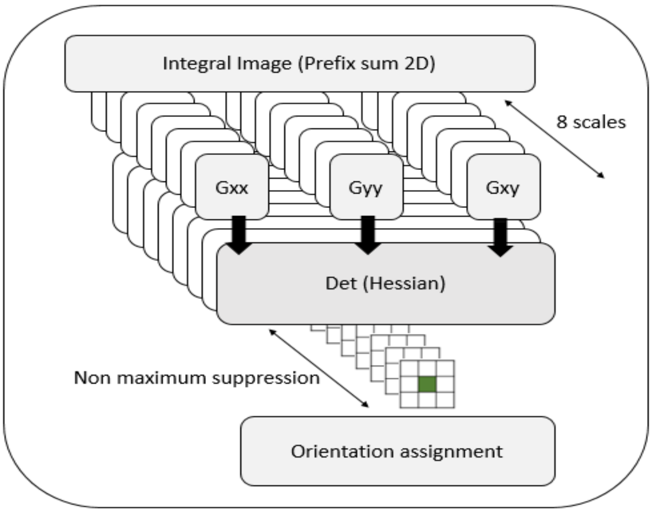

Fig. 5: Parallel steps of SURF

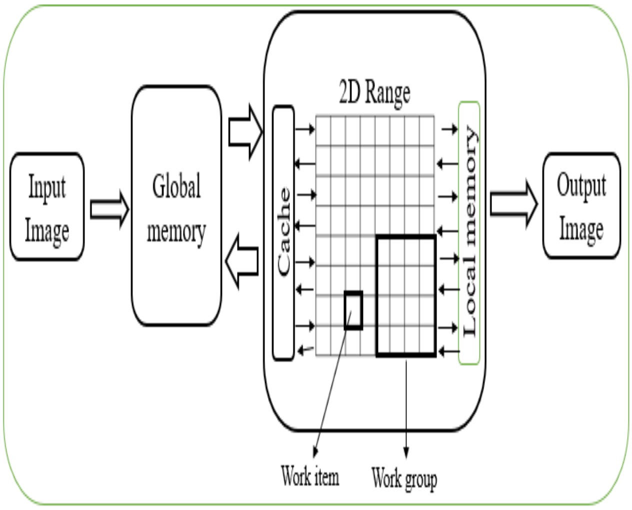

Fig.6: OpenCl Framework for image processing

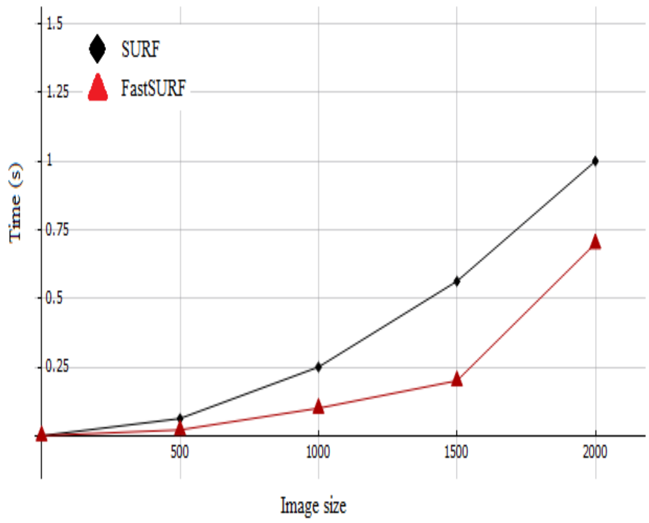

Fig.7: Running time with Original SURF, and FastSURF

Retrieval Number: G5855059720/2020©BEIESP
DOI: 10.35940/ijitee.G5855.069820
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

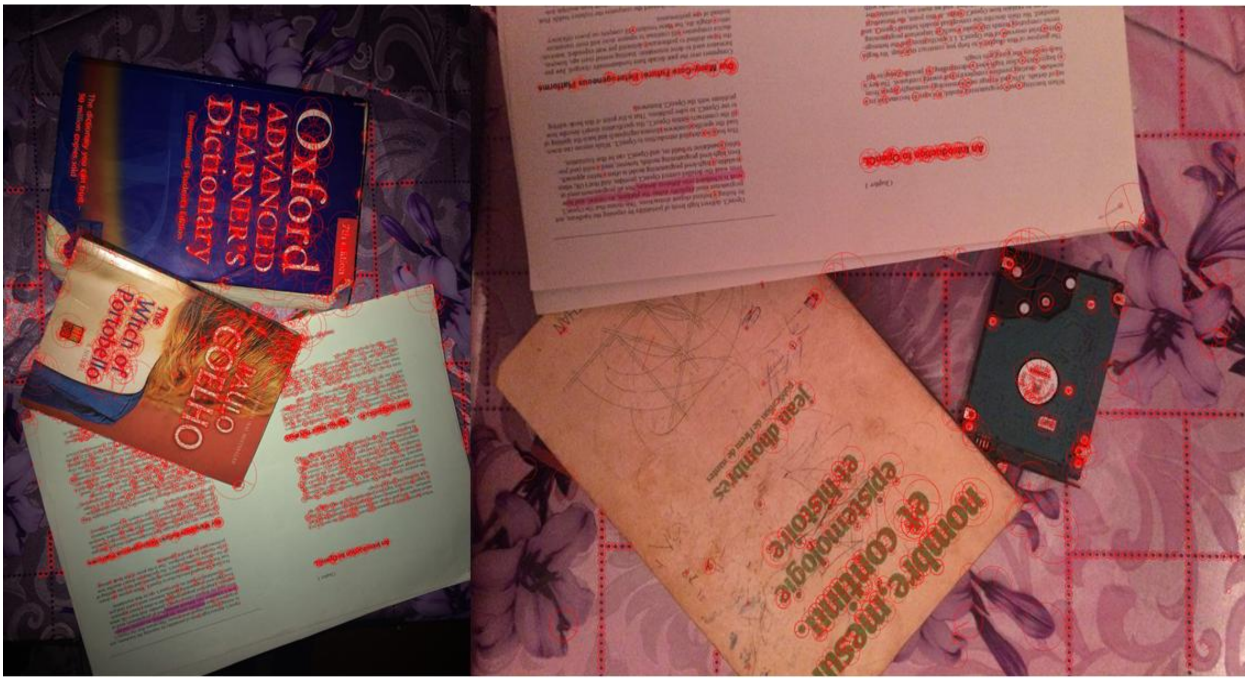

Fig.7: Fast_SURF detector, OpenCl implementation, applied on 2 images of size 3264x2448 with Hessian threshold equal to 200

V. CONCLUSION AND OUTLOOK

In this work we presented a parallel implementation of SURF algorithm on GPU using OpenCl. This technic aims at finding, rapidly, keypoints in a given image. Hense we can resolve a real time problem, with applications such as feature matching and tracking. SURF is chosen because of its robustness, scale and rotation invariance and it is fast and performant. In fact, we combined several methods found in the literature together with our new parallel technic. As a result, we came to the conclusion that we really can reduce greatly computation time.

To improve our algorithm, we are working on:

- Mobile platform implementation using OpenCl.
- Combination of FastSurf and HDR imaging for tracking.

REFERENCES

1. C. G. Harris and M. Stephens, "A combined corner and edge detector.," in Alvey vision conference, 1988, vol. 15, pp. 10–5244.
2. J. Chao, R. Huitl, E. Steinbach, and D. Schroeder, "A novel rate control framework for SIFT/SURF feature preservation in H. 264/AVC video compression," IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 6, pp. 958–972, 2014.
3. C. Wilson et al., "A power-efficient real-time architecture for SURF feature extraction," in 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14), 2014, pp. 1–8.
4. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.
5. T. Lindeberg, "Feature detection with automatic scale selection," International journal of computer vision, vol. 30, no. 2, pp. 79–116, 1998.
6. W. Chen et al., "FPGA-based parallel implementation of SURF algorithm," in 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), 2016, pp. 308–315.
7. T. B. Terriberry, L. M. French, and J. Helmsen, "GPU accelerating speeded-up robust features," in Proceedings of 3DPVT, 2008, vol. 8, pp. 355–362.
8. X. Fan, C. Wu, W. Cao, X. Zhou, S. Wang, and L. Wang, "Implementation of high-performance hardware architecture of OpenSURF algorithm on FPGA," in 2013 International Conference on Field-Programmable Technology (FPT), 2013, pp. 152–159.
9. K. Mikolajczyk, "Interest point detection invariant to affine transformations," PhD Thesis, PhD thesis, Institut National Polytechnique de Grenoble, 2002.
10. M. Brown and D. G. Lowe, "Invariant features from interest point groups.," in BMVC, 2002, vol. 4.
11. D. G. Lowe, "Local feature view clustering for 3D object recognition," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, vol. 1, pp. I–I.
12. D. G. Lowe, "Object recognition from local scale-invariant features," in Proceedings of the seventh IEEE international conference on computer vision, 1999, vol. 2, pp. 1150–1157.
13. H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover.," Stanford Univ Ca Dept of Computer Science, 1980.
14. T. Goldberg and U. Zwick, "Optimal deterministic approximate parallel prefix sums and their applications," in Proceedings Third Israel Symposium on the Theory of Computing and Systems, 1995, pp. 220–228.
15. Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 2004, vol. 2, pp. II–II.
16. T. Lindeberg and L. Bretzner, "Real-time scale selection in hybrid multi-scale representations," in International Conference on Scale-Space Theories in Computer Vision, 2003, pp. 148–163.
17. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in European conference on computer vision, 2006, pp. 404–417

AUTHORS PROFILE

Hicham HASSNAOUI is a PhD student in Computer vision in the Faculty of Sciences and Techniques of Mohammedia (FSTM) of UH2C (University Hassan II - Casablanca) – Morocco. Member of the laboratory EEA&TI (Electronics, Energy, Automatic and Information processing). The research works of H. HASSNAOUI concern Features matching algorithms and HDR Rendering in real time. E-mail: hicham.hassnaoui@gmail.com
Phone: 00212651220660

Aicha SAHEL is a holder of a doctorate in Electronics and Image Processing in 1996 at the University of Poitiers - France. She is University Professor Habilité (Professeur habilité) at FSTM (Faculty of Sciences and Techniques of Mohammedia), of UH2C (University Hassan II - Casablanca) – Morocco, where he teaches the electronics, the signal processing, the image processing and Telecommunications. She is a member of the laboratory EEA&TI (Electronics, Energy, Automatic and Information processing).
The research works of A. SAHEL concern the Communication and Information Technology (Electronics Systems, Signal/Image Processing and Telecommunications). She supervises doctoral theses and she is a co-author of several national and international publications. She is a member in financed research projects. She was a member of steering committees of three international congresses in the same domain of research.
E-mail : sahel_ai@yahoo.fr

Abdelmajid BADRI University of Hassan II of Casablanca · Electrical Engineering Department
19.19 · Prof. HDR (France -1996): Doctorat d'Etat +PhD Doctorat d'Université (1992- France), Qualif. CNU(61°: Génie Informaique, Automatique et Traitement du Signal. Currently works at the Electrical Engineering Department, University of Hassan II of Casablanca. Abdelmajid does research in Electronic Engineering, Computer Engineering and Communication Engineering. Their current project is 'connected objects big data'. E-mail: abdelmajid_badri@yahoo.fr